

Ćwiczenia 8 – biblioteka swing, JFrame, Jpanel, radiobox, checkbox, combobox

Na koniec zajęć prześlij pliki źródłowe i z danymi, wynikami do zasobu w teams.

1. Utwórz nowy projekt w katalogu na dysku C:
2. Efekt końcowy ćwiczeń:



3. Użyte w ćwiczeniach biblioteki: (zostaną zaimportowane automatycznie)

```

*
* @author Andrzej Gac <andrzej.gac@zsmeie.torun.pl>
*/
import java.awt.BorderLayout;
import java.awt.Color;
import java.awt.Container;
import java.awt.Dimension;
import java.awt.Font;
import java.awt.HeadlessException;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.awt.event.KeyEvent;
import javax.swing.JCheckBox;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.JRadioButton;
import javax.swing.JTextArea;
import javax.swing.JToggleButton;

```

4. Zadanie 1: Dodaj nową klasę o nazwie Gui, która dziedziczy po JFrame:

```
public class Gui extends JFrame  
|
```

5. Dodaj konstruktor i ustaw tytuł okna, rozmiar oraz jego widoczność:

```
public Gui() throws HeadlessException {  
    setTitle( title: "-----BorderLayout-----");  
    setSize(new Dimension( width:800, height:500));  
}
```

6. Dodaj widoczność i zamknięcie okna x:

```
    setVisible( b:true);  
    setDefaultCloseOperation( operation:3);  
} // koniec konstruktora
```

7. Zadanie 2: utwórz obiekt klasy Gui:

```
public class Main {  
  
    public static void main(String[] args) {  
        Gui gui = new Gui();  
    }  
}
```

8. Zadanie 3:dodaj kontener, układ border oraz kolor tła:

```
Container c = getContentPane();  
c.setLayout(new BorderLayout());  
c.setBackground( c:Color.CYAN);
```

9. Zadanie 4: Zadeklaruj podstawowe komponenty do testowania:

```
JComboBox lista = null;  
JRadioButton radio = null;  
JCheckBox jCheckBox=null;  
JToggleButton jToggleButton = null;  
JTextArea area=null;  
JPanel jPanelEast =null;  
JPanel jPanelWest =null;  
JPanel jPanelNorth =null;  
JPanel jPanelSouth =null;  
JPanel jPanelCenter =null;
```

10. Zadanie 5: W klasie Gui zdefiniuj panele:

```
jPanelEast = new JPanel();
jPanelWest = new JPanel();
jPanelSouth = new JPanel();
jPanelNorth = new JPanel();
jPanelCenter = new JPanel();
```

11. Zadanie 6: W klasie Gui zdefiniuj formatki:

```
// ---- definicje
area=new JTextArea( rows:25, columns:30);
radio = new JRadioButton( text:"zaznacz, jeśli idziesz do kina");
radio.setMnemonic( mnemonic:KeyEvent.VK_P);
radio.setActionCommand( actionCommand:"start");
jToggleButton = new JToggleButton( text:"OFF");

area.setBackground( bg:Color.LIGHT_GRAY);
area.setFont(new Font( name:"Arial", style:Font.BOLD, size:10));
lista = new JComboBox( items:owoce);
jCheckBox = new JCheckBox( text:"jesteś zapalonym graczem");
jCheckBox.setMnemonic( mnemonic:KeyEvent.VK_C); // lewy Alt+C
jCheckBox.setSelected( b:true);
```

12. Zadanie 7: Ustaw kolory paneli:

```
jPanelEast.setBackground( bg:Color.YELLOW);
jPanelWest.setBackground( bg:Color.GREEN);
jPanelNorth.setBackground( bg:Color.BLUE);
jPanelSouth.setBackground( bg:Color.RED);
```

13. Zadanie 8: Dodaj panele do kontenera w układzie BorderLayout:

```
c.add( comp:jPanelEast, constraints:BorderLayout.EAST);
c.add( comp:jPanelWest, constraints:BorderLayout.WEST);
c.add( comp:jPanelNorth, constraints:BorderLayout.NORTH);
c.add( comp:jPanelSouth, constraints:BorderLayout.SOUTH);
c.add( comp:jPanelCenter, constraints:BorderLayout.CENTER);
```

14. Zadanie 8: Dodaj wszystkie formatki do odpowiednich paneli:

```
jPanelEast.add( comp:lista);
jPanelCenter.add( comp:area);
jPanelWest.add( comp:radio);
jPanelSouth.add( comp:jCheckBox);
jPanelNorth.add( comp:jToggleButton);
```

15. Dodaj obsługę słuchaczy oraz zaimplementuj sugerowane metody:

```
public class Gui extends JFrame implements ItemListener, ActionListener{
```

16. Zadanie 9: Zaimplementuj metodę addActionListener dla jToggleButton:

```
jToggleButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        if (jToggleButton.isSelected()==true){
            area.setText( t: "");
            jToggleButton.setText( text: "ON");
        }
        else {
            area.setBackground( bg: Color.ORANGE);
            jToggleButton.setText( text: "OFF");
        }
    }
});
```

17. Zadanie 10: Dodaj obsługę kliknięcia radiobox:

```
radio.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        if (radio.isSelected() == true) {
            area.append( str: "radio zaznaczone \n");
            System.out.println( x: "tak \n");
        } else {
            area.append( str: "radio nie \n");
        }
    }
});
```

18. Zadanie 11: Dodaj obsługę checkboxa:

```
jCheckBox.addItemListener( this );
```

```
@Override
public void itemStateChanged(ItemEvent e) {
    if (e.getStateChange() == ItemEvent.SELECTED) {
        area.append( str: "tak \n");
        System.out.println( x: "tak \n");
    } else {
        area.append( str: "nie \n");
    }
}
```

19. Zadanie 12: Dodaj metodę addActionListener dla listy rozwijanej:

```
lista.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
  
        area.append(lista.getSelectedItem()+" "  
                    +lista.getSelectedIndex()+"\n");  
    }  
});
```

20. Uruchomienie:



21. Cały kod klasy Gui znajduje się na githubie: <https://github.com/gacandrzej/BorderLayoutNetbeans>

22. KONIEC.